Towards Enabling Exascale Simulation:

SLAC CS/AM Activities for Parallel Finite-Element Electromagnetic Computations

Lie-Quan Lee, Volkan Akcelik, Arno Candel,
Lixin Ge and Greg Schussman

Advanced Computations Department
SLAC National Accelerator Laboratory

The Role of Computational Science

SciDAC Accelerator Projects

Solve the most challenging **computational** problems in accelerator design, optimization and analysis

CS/AM advances supported by CETs and SAP enable **better** and **bigger** simulation

LHC/LARP, Project-X, SuperB, ILC, LCLS, PEP-X, CLIC, Muon Collider, SRF cavities, High-gradient R&D

Linear solver, Eigensolver
Shape optimization
Uncertainty quantification
Adaptive refinement
Dynamic load balancing
Visualization

Accelerator Science & Development

Computational Science

SciDAC
Scientific Discovery through Advanced Computing

Overview

- Uncertainty Quantification of Cavity Shape (TOPS)
  - CEBAF superconducting cavity
- Shape Optimization for Accelerating Cavity (TOPS)
  - Choke cavity for high-gradient concept
- Parallel Domain Specific Linear Solvers (TOPS, CScADS)
  - Linear solvers for saddle-point problems
  - Scalable multilevel preconditioner
  - Out-of-core sparse linear solver
- Novel Algorithms for Solving Large-scale Nonlinear Eigenvalue Problems (NEP) (TOPS, UCDavis)
- Parallel Adaptive Refinement (ITAPS)
- Dynamic Load Balancing (CSCAPES, ITAPS)
- Visualization (IUSV)

Each item can be expanded to a full talk!

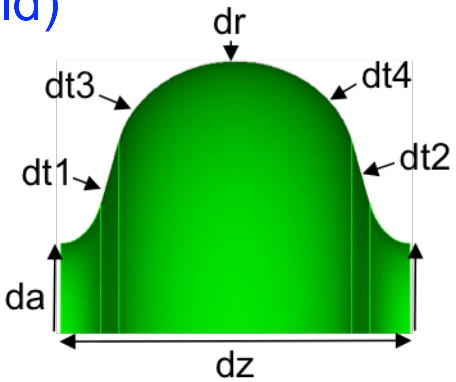
Uncertainty Quantification of Accelerator Cavity Shape

Uncertainty Quantification (UQ) of Cavity Shape

Solve an **inverse problem** to determine the **deformed cavity shape**

- Use measured rf parameters such as f, Q_{ext}, and field profile as inputs
- Parameterize shape deviations using pre-defined geometry variations
- Objective (function J) - minimize weighted least square misfit of the computed and measured responses (f, Q_{ext} and field)

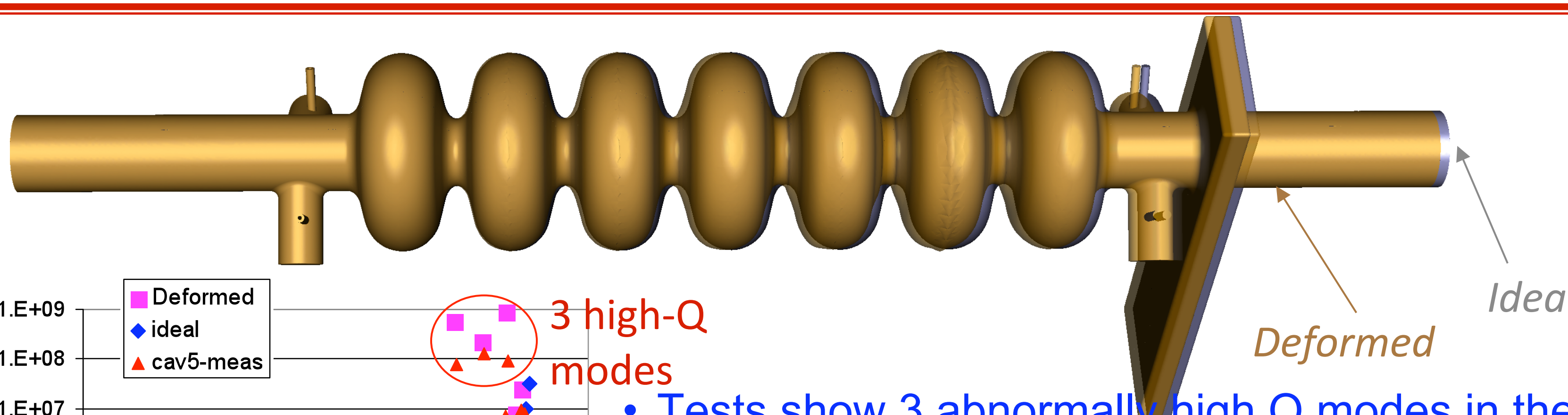
$$\begin{aligned} \underset{\mathbf{e}_j, k_j, \mathbf{d}}{\text{minimize}} \quad & \sum_i \alpha \left(f_i - \bar{f}_i\right)^2 + \sum_i \beta \left(Q_i - \bar{Q}_i\right)^2 \\ \text{subject to} \quad & \mathbf{K}\mathbf{e}_j + ik_j\mathbf{W}\mathbf{e}_j - k_j^2\mathbf{M}\mathbf{e}_j = \mathbf{0} \\ & \mathbf{e}_j^T\mathbf{M}\mathbf{e}_j = 1 \end{aligned}$$

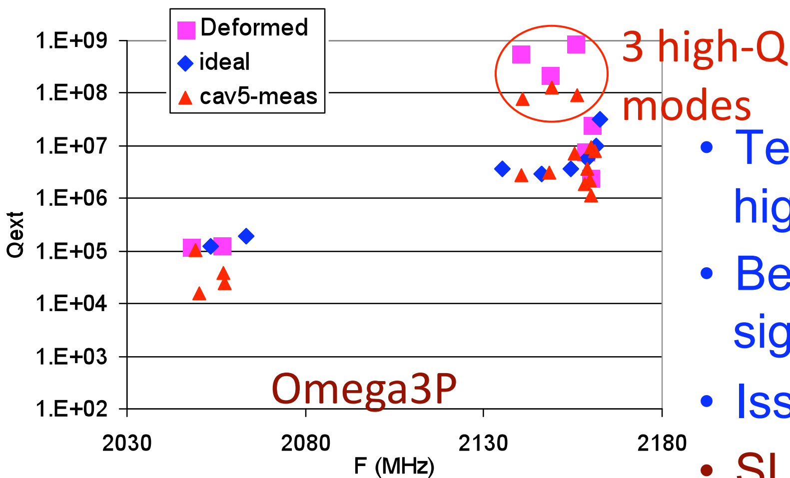


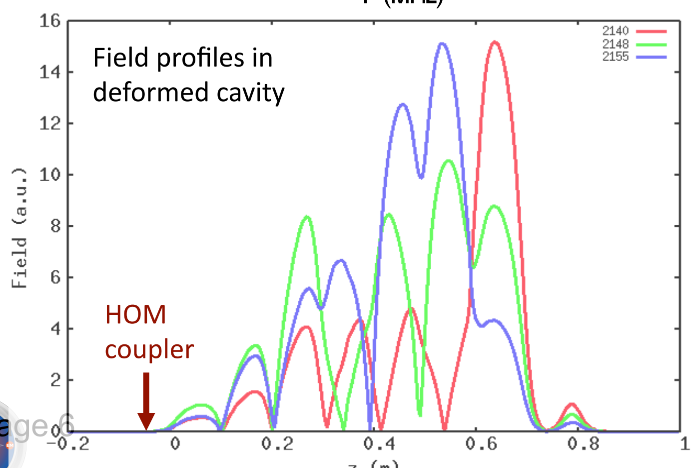
- Regularization or truncated SVD employed to deal with noisy data
- The optimization procedure typically converges within a handful of nonlinear iterations with Newton type algorithms

Ref: V. Akcelik et al., "Shape Determination for Deformed Electromagnetic Cavities", J. Comput. Phys., **227**, 1722 (2008).

UQ for CEBAF BBU: Simulation & Analysis







- Tests show 3 abnormally high Q modes in the high-gradient cavities
- Beam-breakup (BBU) threshold current is significantly below designed value
- Issues could not be resolved experimentally
- <u>SLAC has made great progress by treating it as an inverse problem</u>
- Identified the main cause of the BBU instability: **Cavity is 8 mm shorter** – this is confirmed later from measurements
- Success requires **a multidisciplinary effort** in accelerator modeling, computational science and RF measurements

Shape Optimization for Cavity Design

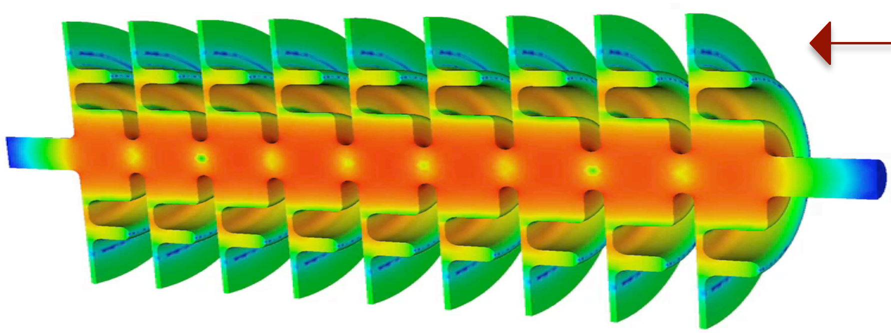
Design of Choke Cavity

Design goals:

- Set accelerating mode frequency to 11.424 GHz.
- Satisfy field flatness for the accelerating mode.
- Maximize external Q for the accelerating mode.
- Minimize external Q value for the higher order modes (HOM).
- Constrain the voltage for the accelerating mode.

Shape parameters:

- Design variables are CAD parameters.
- 21 design parameters, 7 middle cells need to be identical.
- Design parameters have simple bounds.



Example of non-optimized cavity: accelerating mode leaks through choke

Shape Optimization for Choke Cavity

Optimization Problem:

$$\underset{\mathbf{e}_j, k_j, \mathbf{d}}{\text{minimize}} \quad -\beta Q_a + \alpha \sum Q_{HOM} + \gamma \sum_{j=1}^{9} (|\mathbf{e}(\mathbf{x}_i)| - |\bar{\mathbf{e}}|)^2 - \delta V_a$$

$$\text{subject to} \quad \mathbf{K}\mathbf{e}_j + i k_j \mathbf{W} e_j - k_j^2 \mathbf{M}\mathbf{e}_j = \mathbf{0}$$

$$\mathbf{e}_j^T \mathbf{M}\mathbf{e}_j = 1$$

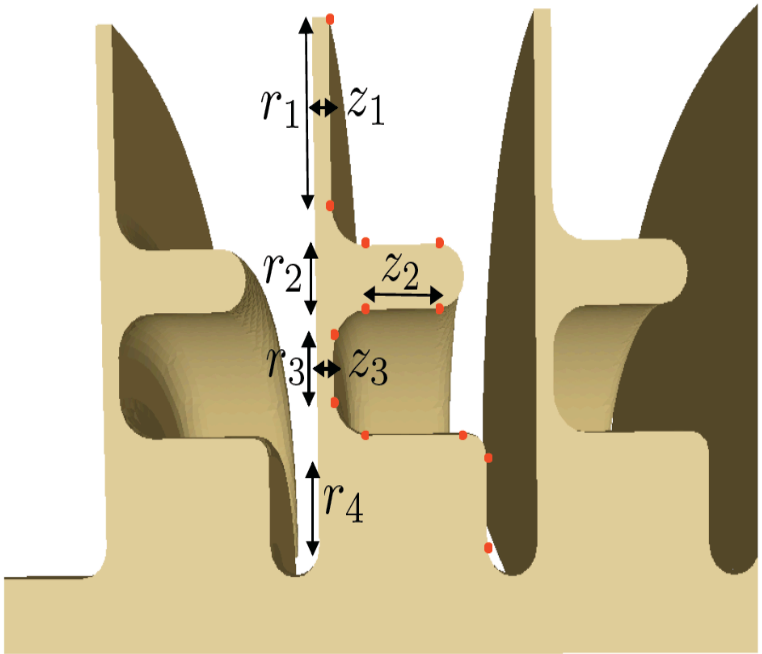
$$f_a = 11.424e9$$

$$l_i <= d_i <= u_i$$

Resulting Shape:

Optimized parameters in μm						
Cell 1						
r1	r2	r3	r4	z1	z2	z3
117.3	-727.7	-74.9	-27.34	349	186.1	743.9
Cell 2-8						
r1	r2	r3	r4	z1	z2	z3
604.5	-1754.8	1178.1	-135.3	1800	651.5	36.1
Cell 9						
r1	r2	r3	r4	z1	z2	z3
15.6	-238.4	-42.7	32.3	744	96.2	-347.5

Design Parameters



Movie for Choke Cavity Design

Domain Specific Scalable Linear Solver

Linear Solver for Saddle-point Problems

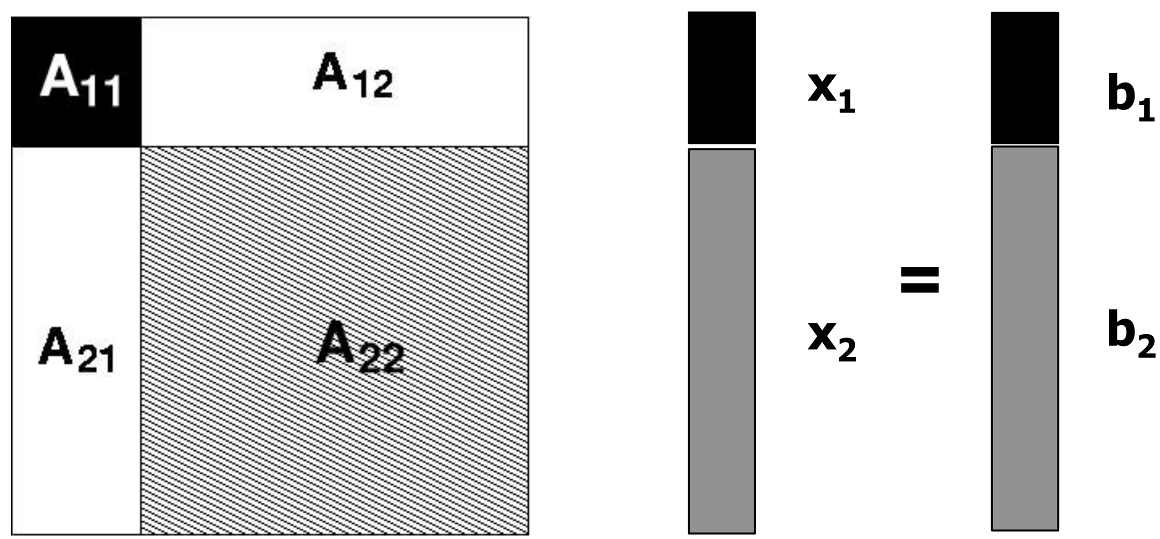
- Solving linear systems from Karush-Kuhn-Tucker (KKT) condition is one of the keys in solving shape optimization and uncertainty quantification

$$\begin{pmatrix} \mathbf{K} - \lambda \mathbf{M} & \mathbf{Mv} \\ (\mathbf{Mv})^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \xi \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}$$

- It is highly indefinite – both (1,1) and (2,2) block are singular: (λ, v) satisfies Kv = λMv
- Direct algorithm
  - A sparse direct solver is applied to (0,0) block
  - Null space is removed through orthogonalization
- Iterative algorithm for (K-λM) t = b - ξ Mv
  - Remove null space from solution of precondtioning system

Multilevel Preconditioner

- Matrices from high-order finite-element simulation can be partitioned into two-by-two blocks
- A multilevel preconditioner: A_{11} can factorized while A_{22} can be approximated (IPDPS05 and CSE07)
- Advantage: convergence is independent of mesh size
- Problem: A_{11} is too small and scalability of factorization and triangular solver are bad
- Solution: (next page)



Scalable Implementation of Multilevel Preconditioner

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

New Implementation highlights:

- A_{11} is factorized and solved with a subset of MPI processors with threading
  - Triangular solver of factorized A_{11} uses much less wall clock time!
- A_{22} is approximated with incomplete LU factorization with all processors
- Coupling terms makes solver converges fasters

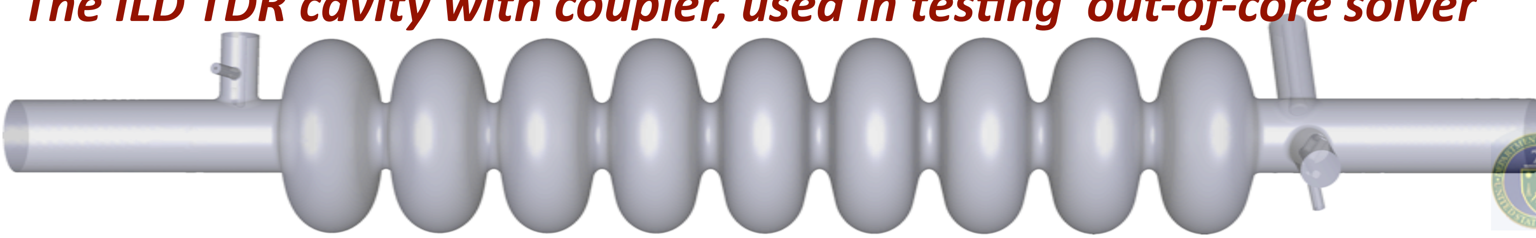
More scalable solver:

- Speed: much less overall wall clock time
- Memory usage: problems cannot be solved on NERSC bassi (per-node memory 32GB) before can be solved on NERSC franklin (per-node memory 8GB) now

Exploring Out-of-Core Solver

- ❑ Available amount of memory limits us to use sparse direct solver for larger problem-size
- ❑ Out-of-core techniques save the matrix factor into disk and use it in triangular solver
- ❑ MUMPS out-of-core solver has been integrated into Omega3P
- ❑ Example: Solving ILC TDR cavity with couplers for first monopole bands
  - ➢ 531k tetrahedral elements, 2nd order finite element bases, 3.1 million DOFs, 4 cores AMD Opteron Processor , 6 hours wall clock time
  - ➢ As a comparison, same problem, on NERSC bassi with 64 CPUs and 256GB memory, 10 minutes
  - ➢ Out-of-core solver will solve larger problem-size by using disk space as temporary memory, but it trades off the execution time

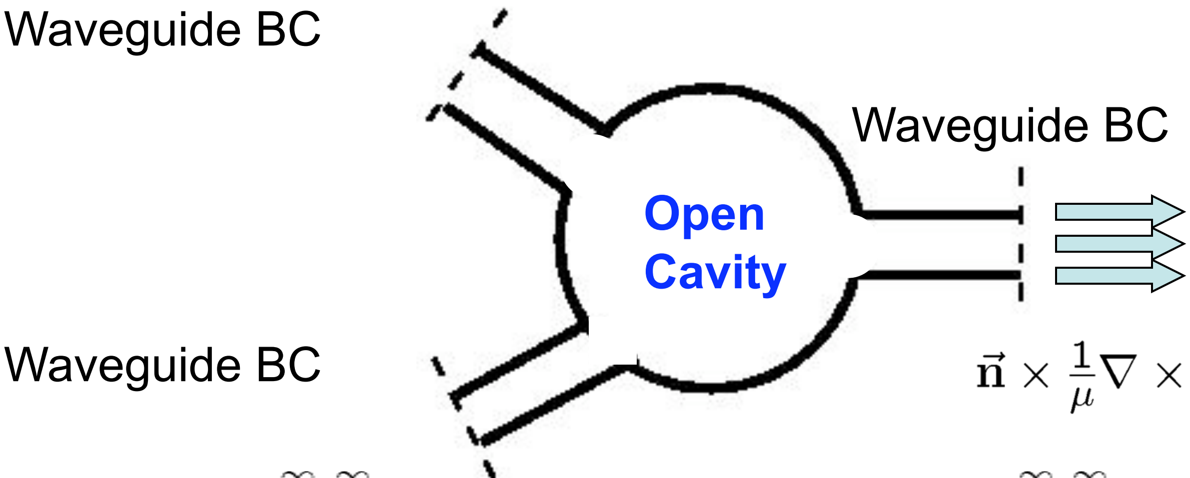
The ILD TDR cavity with coupler, used in testing out-of-core solver



Novel Algorithm for Nonlinear Eigenvalue Problems

Nonlinear Eigenvalue Problems (NEP) in Accelerator Cavity Simulation

Cavity loaded with multiple waveguide modes



$\vec{\mathbf{e}}^{TM}_{tmn}$

$\vec{\mathbf{e}}^{TE}_{mn}$

$\vec{\mathbf{n}} \times \frac{1}{\mu} \nabla \times \vec{\mathbf{E}} + P(\vec{\mathbf{E}}) = 0$ where

$$P(\vec{\mathbf{E}}) = \sum_{m}^{\infty} \sum_{n}^{\infty} \frac{k^2}{i\sqrt{k^2 - k^2_{mn}}} \vec{\mathbf{e}}^{TM}_{tmn} \int_\Gamma \vec{\mathbf{e}}^{TM}_{tmn} \cdot \vec{\mathbf{E}}\, d\Gamma - \sum_{m}^{\infty} \sum_{n}^{\infty} i\sqrt{k^2 - k^2_{mn}}\, \vec{\mathbf{e}}^{TE}_{mn} \int_\Gamma \vec{\mathbf{e}}^{TE}_{mn} \cdot \vec{\mathbf{E}}\, d\Gamma$$

- Vector wave equation with waveguide boundary conditions can be modeled by a nonlinear eigenvalue problem

$$\mathbf{K}x + i \sum_{m,n} \sqrt{k^2 - k^2_{mn}}\, \mathbf{W}^{TE}_{mn} x + i \sum_{m,n} \frac{k^2}{\sqrt{k^2 - k^2_{mn}}} \mathbf{W}^{TM}_{mn} x = k^2 \mathbf{M} x$$

where

$$(\mathbf{W}^{TE}_{mn})_{ij} = \int_\Gamma \vec{\mathbf{e}}^{TE}_{mn} \cdot \mathbf{N}_i\, d\Gamma \int_\Gamma \vec{\mathbf{e}}^{TE}_{mn} \cdot \mathbf{N}_j\, d\Gamma$$
$$(\mathbf{W}^{TM}_{mn})_{ij} = \int_\Gamma \vec{\mathbf{e}}^{TM}_{tmn} \cdot \mathbf{N}_i\, d\Gamma \int_\Gamma \vec{\mathbf{e}}^{TM}_{tmn} \cdot \mathbf{N}_j\, d\Gamma$$

Novel Algorithm for NEP

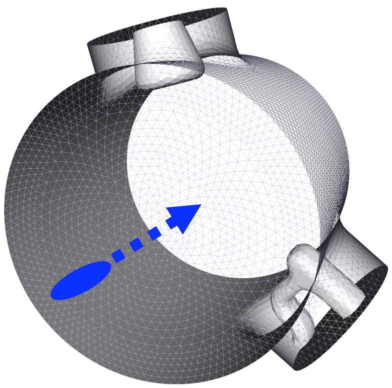
$$\mathbf{K}x + i\sum_{m,n}\sqrt{k^2 - k_{mn}^2}\,\mathbf{W}_{mn}^{TE}x + i\sum_{m,n}\frac{k^2}{\sqrt{k^2-k_{mn}^2}}\mathbf{W}_{mn}^{TM}x = k^2\mathbf{M}x$$

- Nonlinear Jacobi-Davidson algorithm and Self-Consistent Iterations are two primary algorithms for NEP
  - Eigenvectors are not orthogonal
- New algorithm developed by Z. Bai (UC Davis) and LBL scientists (TOPS)
  - Algorithm description
    - ✓ Padé approximation for nonlinear terms
    - ✓ NEP becomes rational eigenvalue problems (REP)
    - ✓ Solve linearized REP
  - Advantage
    - ✓ Many existing algorithms for linear eigenvalue problems (LEP)
    - ✓ The size of LEP is only slightly larger than that of NEP
    - ✓ Much faster overall execution time in preliminary testing

Parallel Adaptive Refinement for Time-Domain Finite-Element Simulation

P-refinement for Short-range Wakefield

ILC short-range wakefield simulation

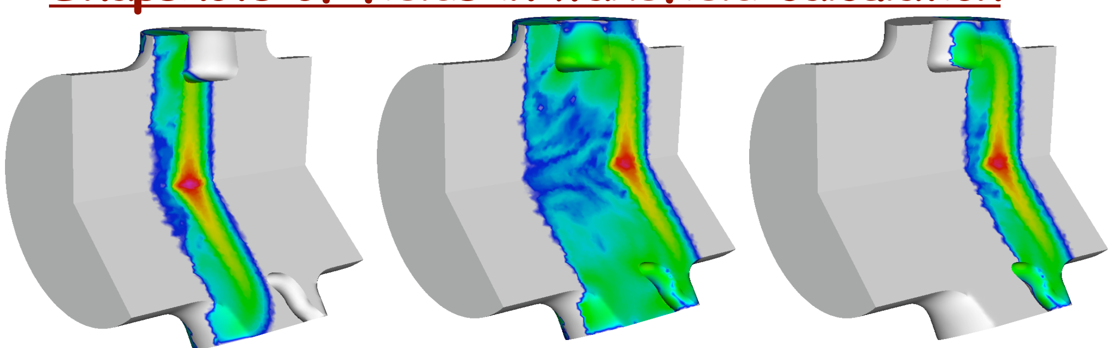


- ❑ Beam size ~ 300 micron
- ❑ Beam pipe radius: 39 mm
- ❑ Estimated > 100 million tetrahedral elements just for coupler!

Moving window with p-refinement

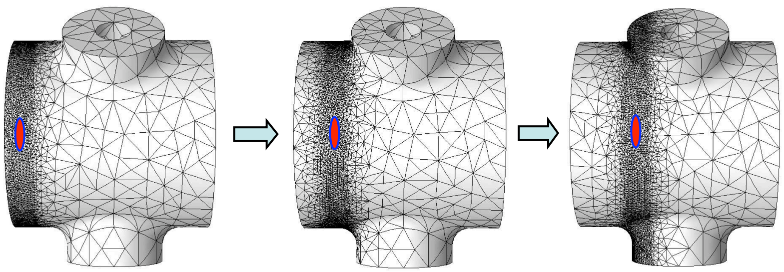
- ❑ Inside window: p > 0
- ❑ Outside window: p = 0
- ❑ Significantly reduces execution time and memory usage

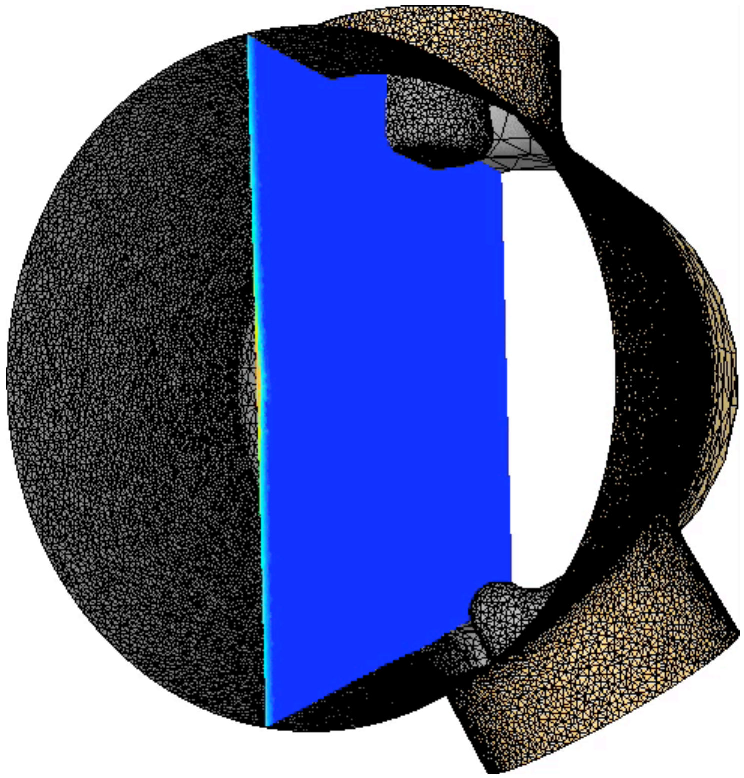
Snapshots of fields in wakefield calculation



- ❑ 800 micron beam size
- ❑ 400 micron edge length
- ❑ 13 million elements
- ❑ 5 windows in the run
- ❑ 1/10th of execution time
- ❑ 1/10th of memory usage

h-refinement for Short-range Wakefield





h-refined moving window

- Refined mesh only around moving beam, thereby
- reducing computational resources by orders of magnitude

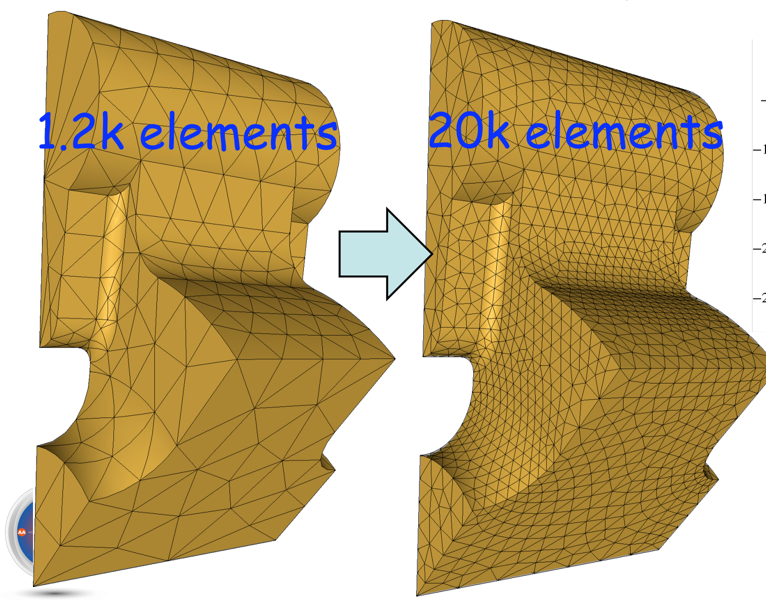
Ref: X. Luo. M. Shephard, L.-Q. Lee, C. Ng, L. Ge, "Tracking Adaptive Mesh Refinement in 3D Curved Domains for Large-Scale Higher-Order Finite-Element Simulations", Best Meshing Technical Poster Award at the 17th International Meshing Roundtable, Pittsburgh, Oct. 12-15, 2008.

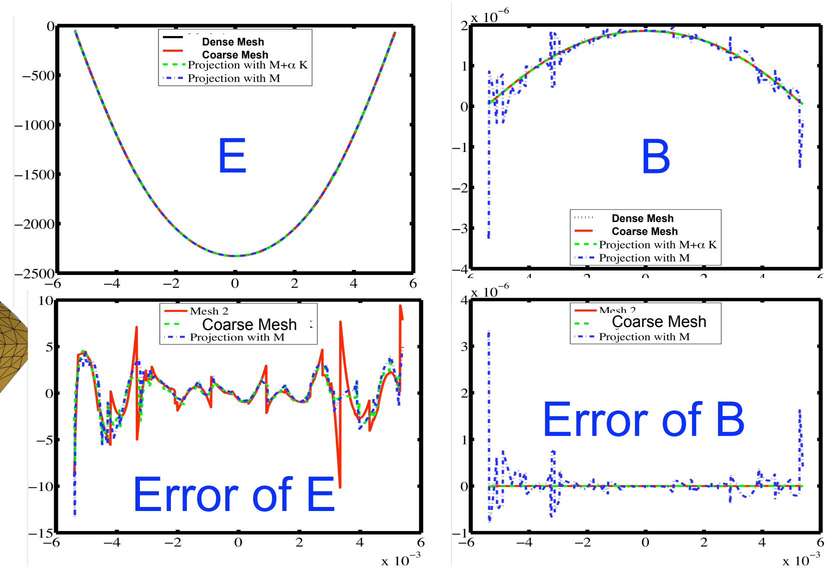
Solution Transfer between Different Meshes

- Discretized electromagnetic fields need to be transferred between different unstructured meshes (h-refinement, mesh-based multi-level preconditioner)
- A new projection method is discovered:

$$(\mathbf{M}+\alpha\mathbf{K})\,\mathbf{x}^b=\left(\vec{\mathbf{N}}_j^b,\varepsilon\sum_i x_i^a\vec{\mathbf{N}}_i^a\right)+\alpha\left(\nabla\times\vec{\mathbf{N}}_j^b,\frac{1}{\mu}\sum_i x_i^a\nabla\times\vec{\mathbf{N}}_i^a\right)$$

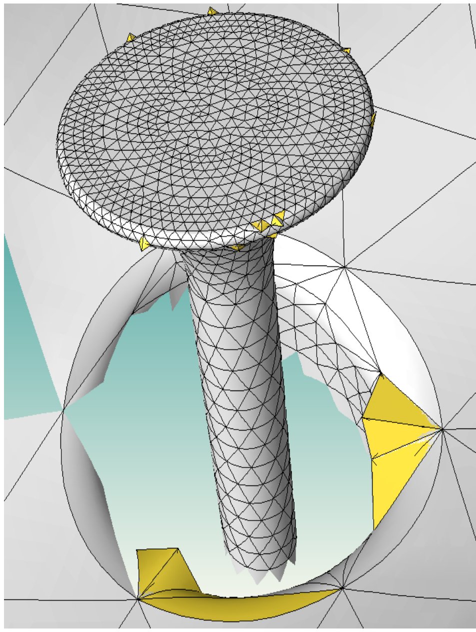
where $\mathbf{M}_{ij}=\left(\varepsilon\vec{\mathbf{N}}_i,\vec{\mathbf{N}}_j\right)$ and $\mathbf{K}_{ij}=\left(\frac{1}{\mu}\nabla\times\vec{\mathbf{N}}_i,\nabla\times\vec{\mathbf{N}}_j\right)$



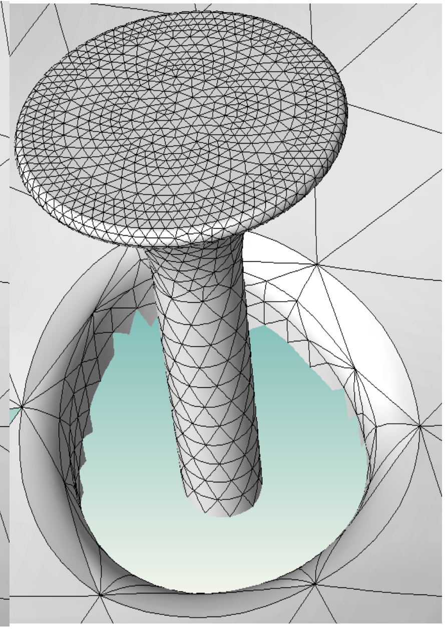


- Balance errors of both E and B
- Keep the quality of the solution

Mesh Curving Correction



Regions with invalidly curved elements



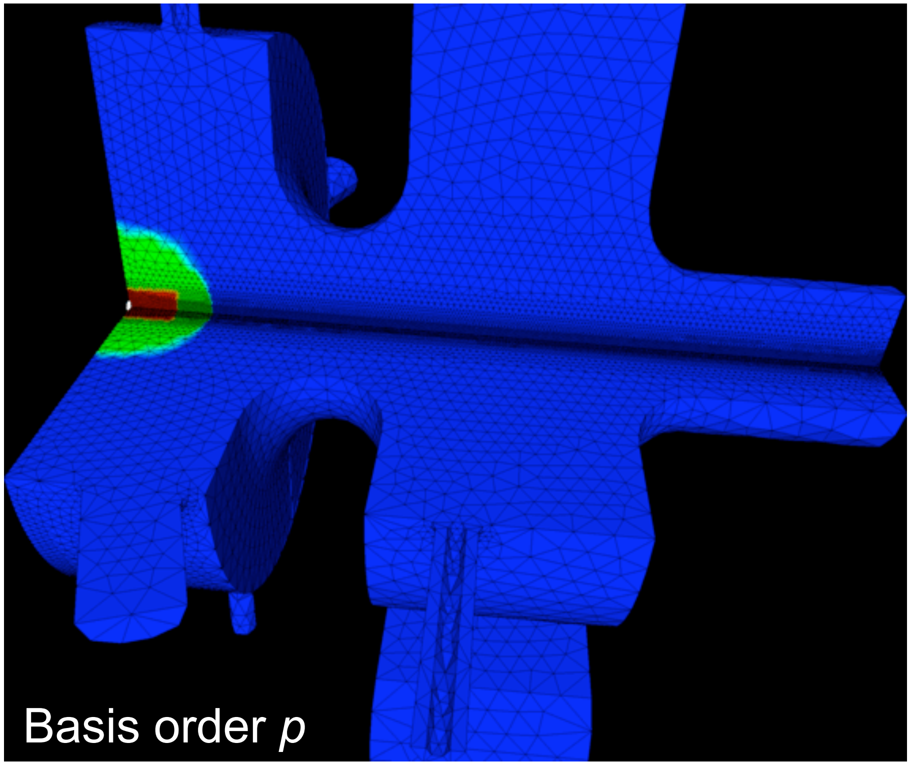
The same region after mesh curving correction

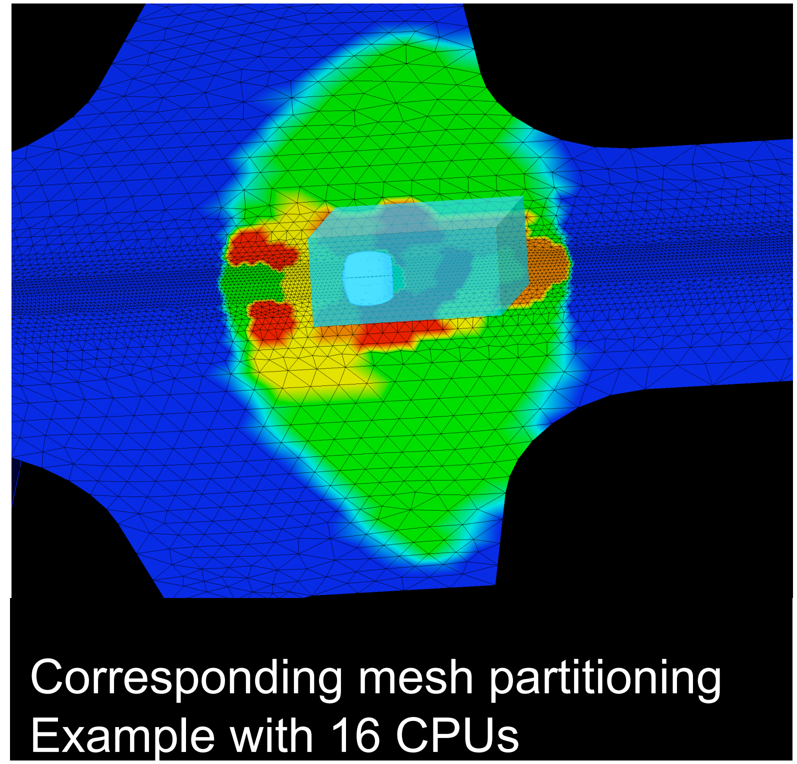
- RPI scientists create a tool
- Corrected mesh cures numerical instability in T3P
- Reduces T3P simulation time drastically
- More in Mark Shephard's presentation later

Also See: Presentation of Walter Polansky, "Scientific Discovery Through Advanced Computing and the Path Toward Computing at Extreme Scale", 2008

Dynamic Load Balancing for PIC Electromagnetic Simulation

Causal Adaptive *p*-refinement for PIC3P





Blue: 0th order	Green: 1st order	Red: 2nd order

Restrict calculations onto causal domain:
0th order means no field calculations…
<u>But particles don't notice any difference!</u>

<u>PIC Domain now significant part of total computational domain:</u>

Need a good particle-field load balancing scheme

New Load Balancing Method

Requirements:

- Strong scalability (same problem runs faster with more CPUs)
- Weak scalability (can solve larger problem with more CPUs)
- Should work near optimal for any particle distribution
- Small overhead for typical cases

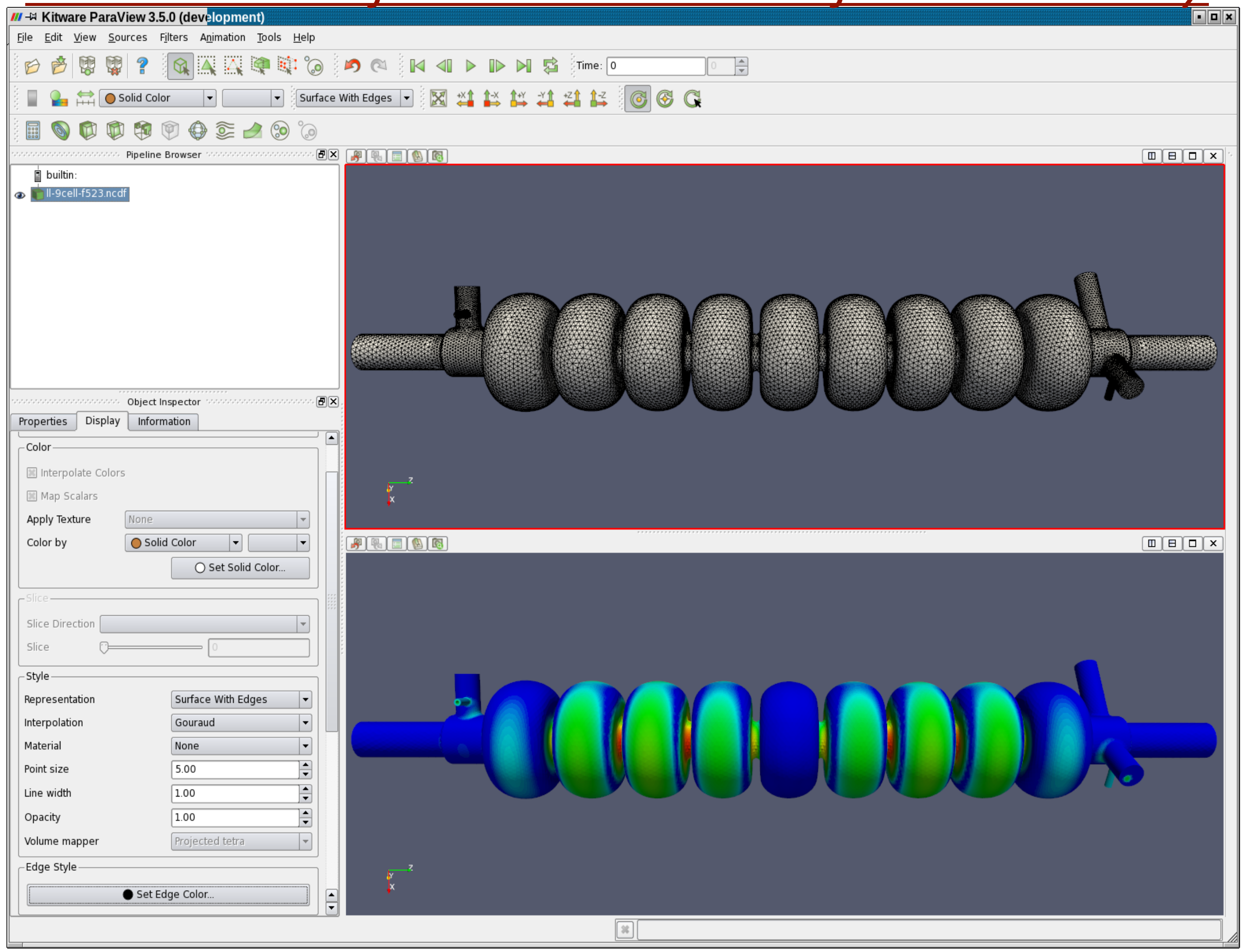
Proposed Solution (implementing now):

- Partition particles (and fields) geometrically (RCB)
- Every CPU owns a compact sub-bunch of particles 
- Every CPU needs fields in particle region
- Every CPU knows whom to get the fields from (and send current to)
- Some-to-some communication (instead of some-to-one-to-all)
- Re-partitioning after every few steps to keep comm. volume low

Parallel Visualization

- SLAC was not funded with parallel visualization through SAP activity
- Parallel visualization is essential to accelerator modeling (25GB per mode for cryomodule, 5.5TB data for 40ns)
- Mesh and mode readers for Paraview (a parallel viz toolset) have been implemented

A Screenshot of mesh and a mode for an ILC cavity



New Collaboration Opportunities

- Including CAD and mesh smoothing in shape optimization and uncertainty quantification
- Including CAD into mesh curving tool
- Multiphysics and multiscale simulation
  - Anisotropic mesh
- Performance optimization and improvement
- Memory-usage scalability of all the computational components in FEM simulation

Thank You